# Energy Overhead of Meltdown and Spectre Mitigations on Linux

GI Fachgruppe Betriebssysteme – Herbsttreffen
September 22, 2021

**Benedict Herzog**[1], Stefan Reif[2], Julian Preis[2], Timo Hönig[1], Wolfgang Schröder-Preikschat[2]

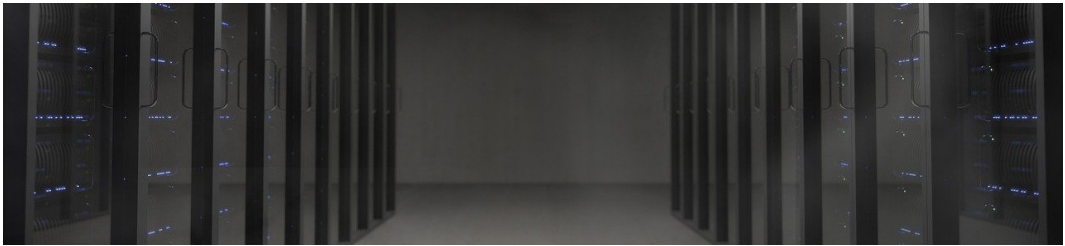
[1] Ruhr-Universität Bochum (RUB)
[2] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

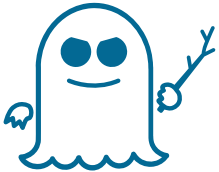
RUHR UNIVERSITÄT BOCHUM

RUB

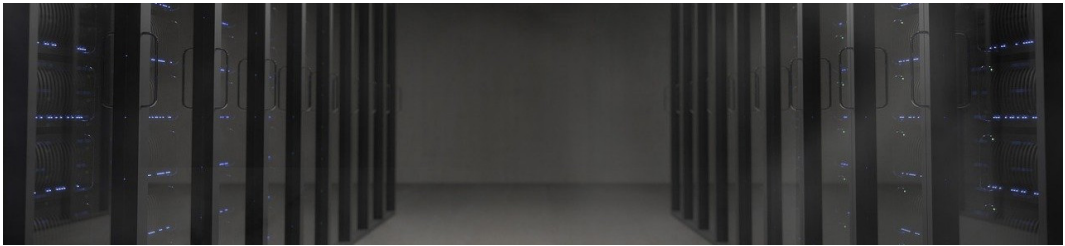
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

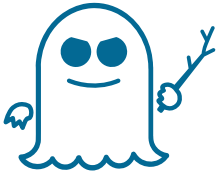
# The Price of Meltdown and Spectre





# The Price of Meltdown and Spectre





# The Price of Meltdown and Spectre (2)

**The goal of this work is to put a price tag on the Meltdown/Spectre software mitigations in terms of their energy overhead[1].**

[1] Herzog et al., **The Price of Meltdown and Spectre: Energy Overhead of Mitigations at Operating System Level**. *Proc. of the 14th European Workshop on Systems Security (EuroSec'21). 2021*

# Research Questions

**Q1** **How much energy overhead** is introduced by Meltdown/Spectre mitigations?

**Q2** Is the energy overhead **related to specific subsystems** (e.g., CPU, block I/O)?

**Q3** Is the energy overhead **correlated with the execution time** overhead?

**Q4** Is the energy overhead **predictable** for a given application?

# Agenda

Motivation

Meltdown and Spectre Mitigations

Energy Overhead Analysis

Energy Overhead Prediction

Conclusion

# Meltdown and Spectre Attacks and Mitigations

## Attacks

- class of hardware vulnerabilities
- (time) side-channel based
- bypass memory access protection

## Mitigations

- full mitigation at hardware-level
- partial mitigation at software-/firmware-level

# Meltdown and Spectre Attacks and Mitigations (2)

**Attacks**

**Mitigations**

**Meltdown:**

→ Linux `[no]pti`

Kernel Page Table Isolation (KPTI)

**Spectre v1:**

→ Linux `[no]spectre_v1`

`swapgs`/usercopy barriers,
pointer sanitization

**Spectre v2:**

→ Linux `[no]spectre_v2`

retpolines,
Indirect Branch Restricted Speculation (IBRS),
Return Stack Buffer (RSB) refilling

# Measurement Methodology

**Benchmarks**

Training/Analysis — Evaluation

| sysbench | stress-ng | Phoronix |
|---|---|---|
| 4 benchmarks | 24 benchmarks | 10 benchmarks |
| 20 s-300 s exec. time | 20 s-300 s exec. time | 2 m-15 m exec. time |
| 10 (50) iterations | 10 (50) iterations | 10 iterations |

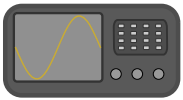
intermingle →

```
$> ./sysbench-1
$> ./stress-ng-1
$> ./sysbench-2
[...]
```

execute ↓

- Intel Core i5 8400 (2.8 GHz)
- 8 GB RAM
- 2 TB HDD
- 1 Gbit Ethernet
- Ubuntu 18.04 LTS
  - `nopti`
  - `nospectre_v1`
  - `nospectre_v2`

`perf` + **Intel RAPL**

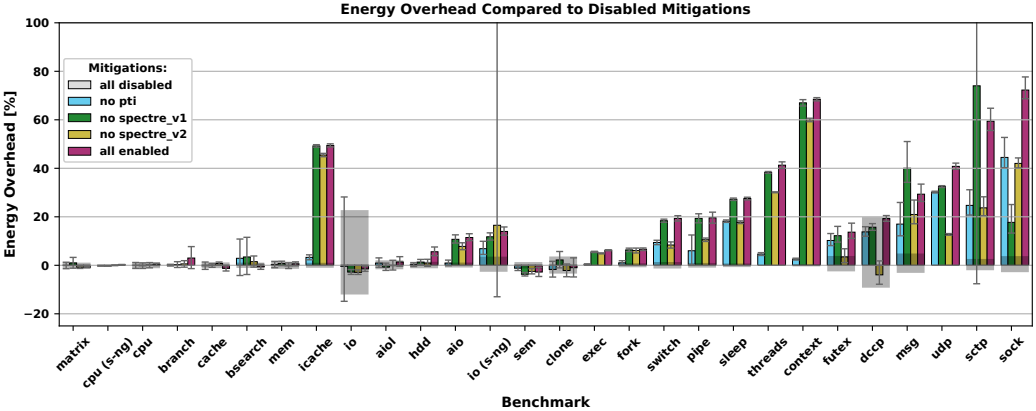


measure energy

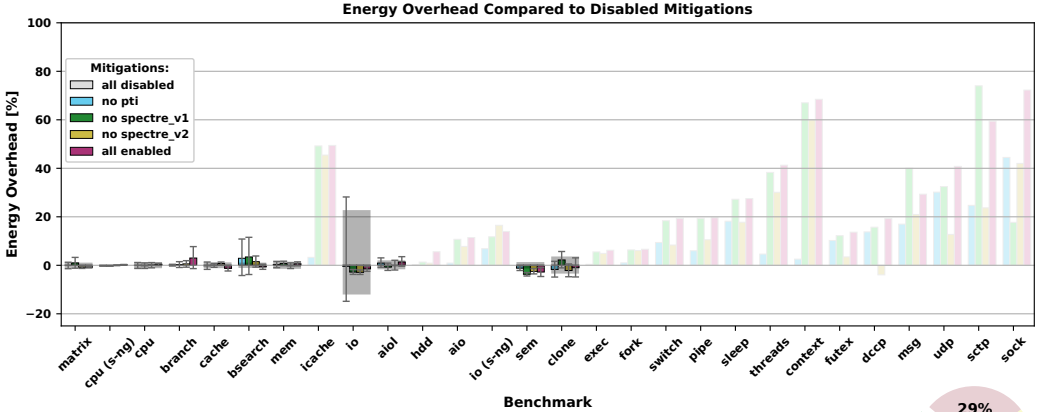
collect performance counters →

- 4 perf. counters
- one iteration
- 1 ms sampling rate
- CPU package

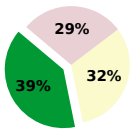
# Q1: Energy Overhead of Spectre/Meltdown



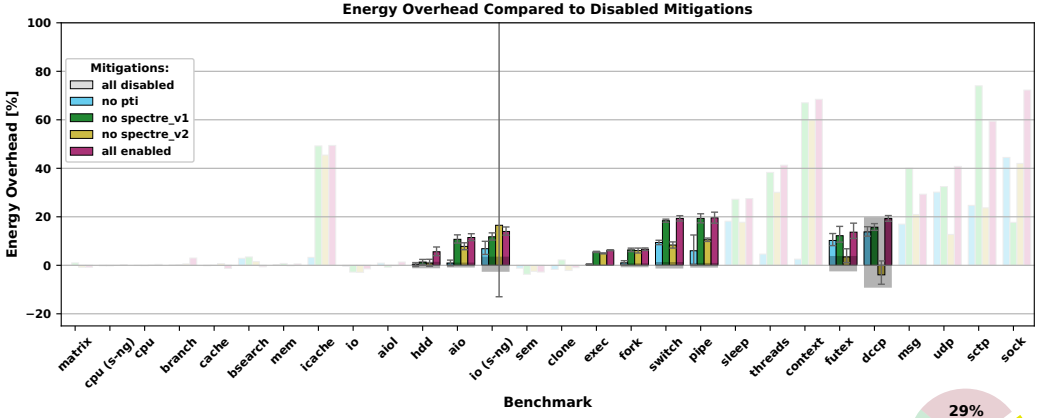
# Q1: Energy Overhead of Spectre/Meltdown



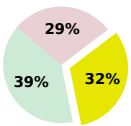

→ **11 out of 28 benchmarks have an overhead below 5 %**



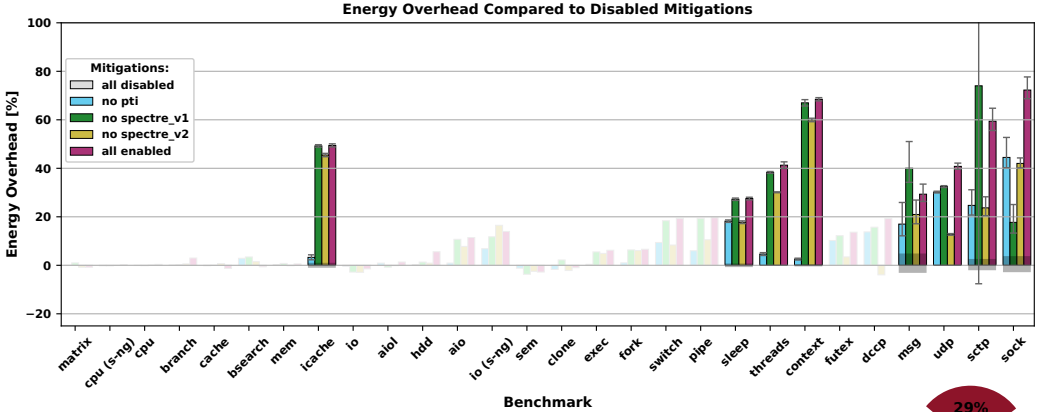
# Q1: Energy Overhead of Spectre/Meltdown



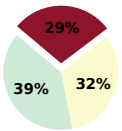

→ **9 out of 28 benchmarks have an overhead between 5 % and 25 %**



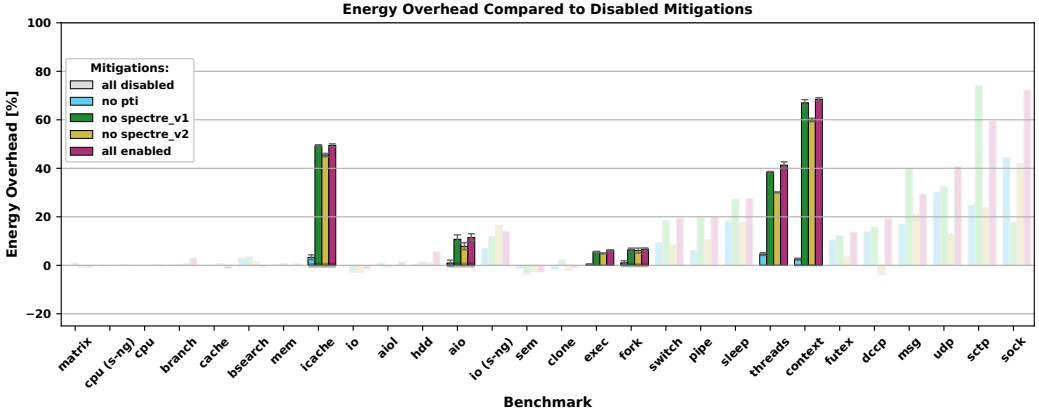
# Q1: Energy Overhead of Spectre/Meltdown




→ **8 out of 28 benchmarks have an overhead above 25 %**

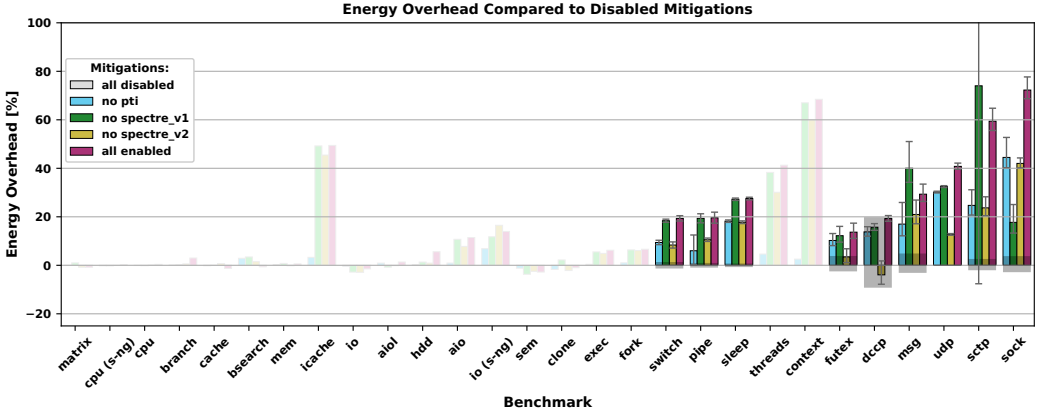


# Q1: Energy Overhead of Spectre/Meltdown



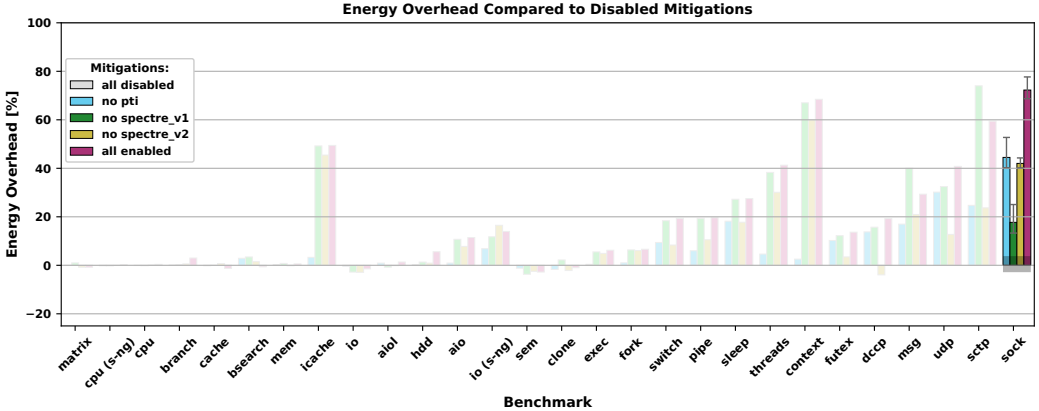

→ **KPTI often has the greatest influence**

# Q1: Energy Overhead of Spectre/Meltdown



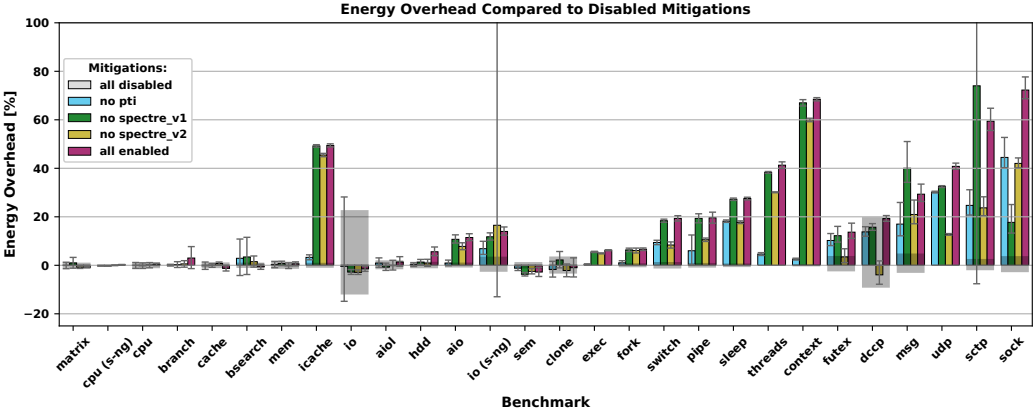

→ **Spectre v2 also contributes to the overhead**

# Q1: Energy Overhead of Spectre/Meltdown



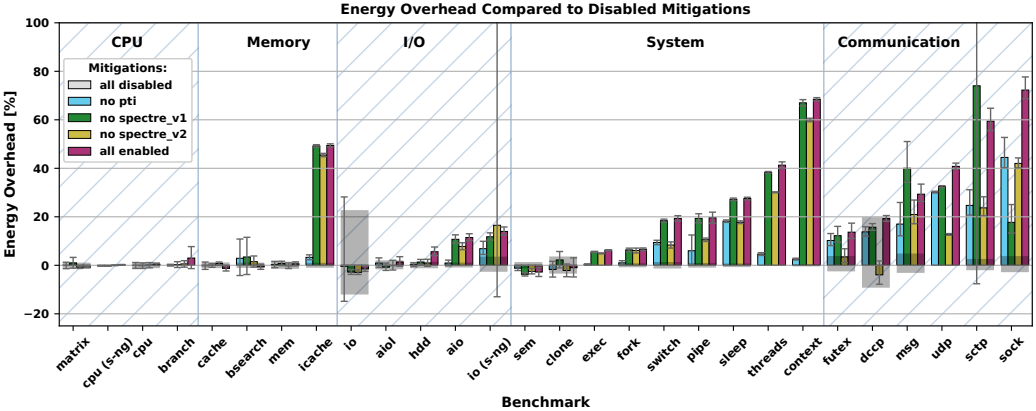

→ **Spectre v1 only influences one benchmark**

# Q1: Energy Overhead of Spectre/Meltdown

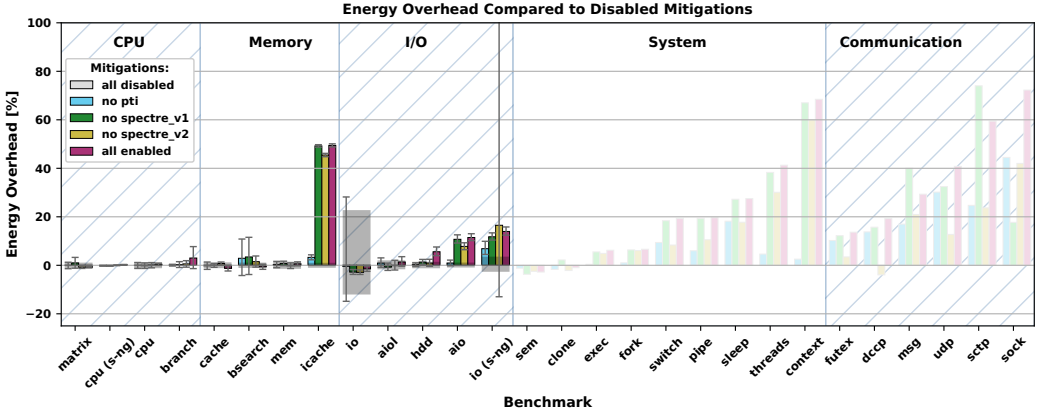



**The overhead is highly application-dependent and lies between ~0 % and 72 %.**

# Q2: Relations between Mitigations' Overhead and Subsystem

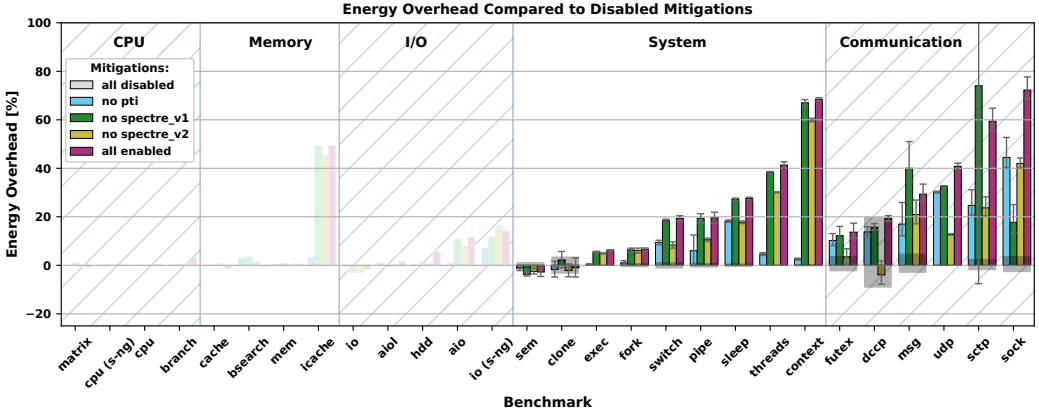


# Q2: Relations between Mitigations' Overhead and Subsystem



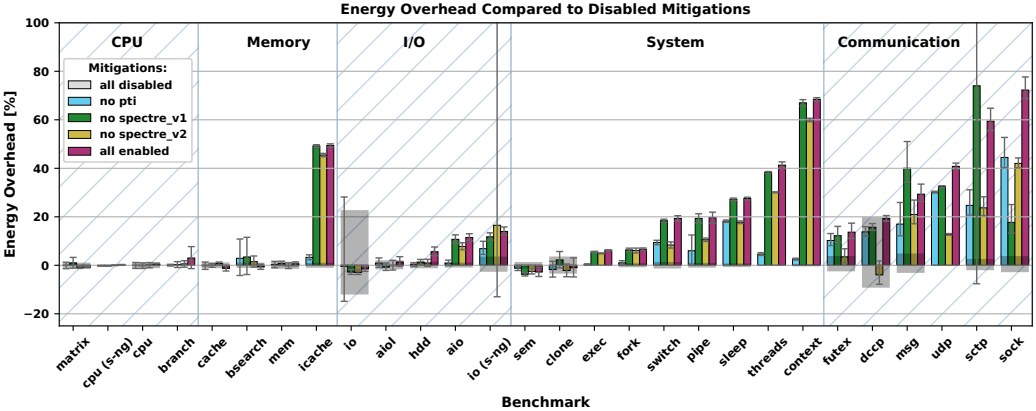

→ **CPU-, memory-, and I/O-heavy benchmarks have (mostly) no or small overheads**

# Q2: Relations between Mitigations' Overhead and Subsystem



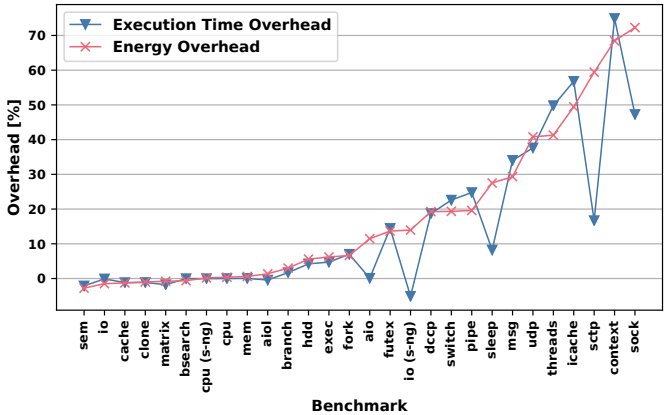

→ System- and communication-heavy benchmarks have in general higher overheads

# Q2: Relations between Mitigations' Overhead and Subsystem



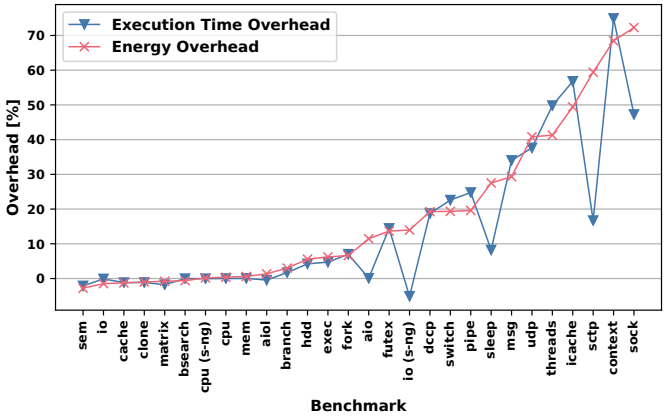

**System interactivity greatly influences the mitigations' overhead**

# Q3: Correlation between Energy and Execution Time Overhead




- positive correlation
- Spearman correlation coefficient: 0.88
- 5 noticeable exceptions

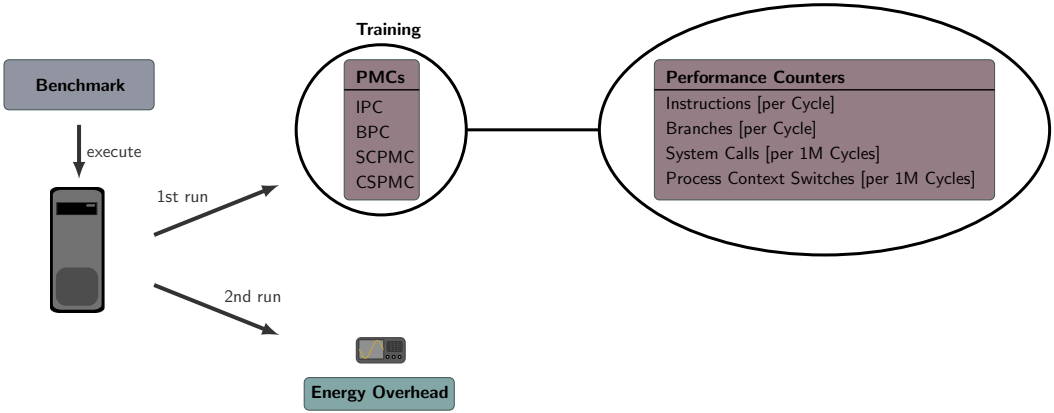
# Q3: Correlation between Energy and Execution Time Overhead



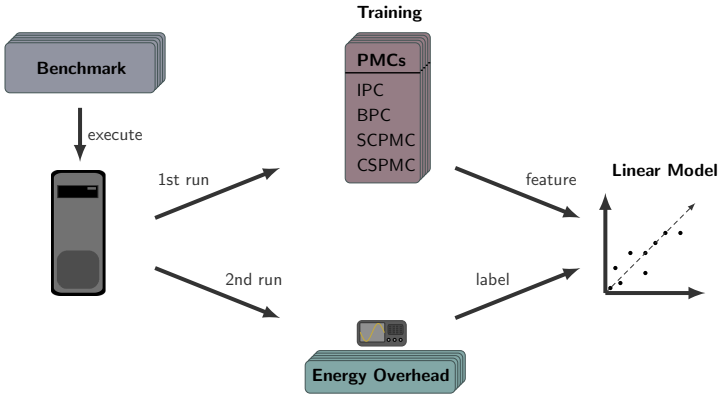

- positive correlation
- Spearman correlation coefficient: 0.88
- 5 noticeable exceptions

**Energy and execution time overhead are correlated (exceptions apply)**

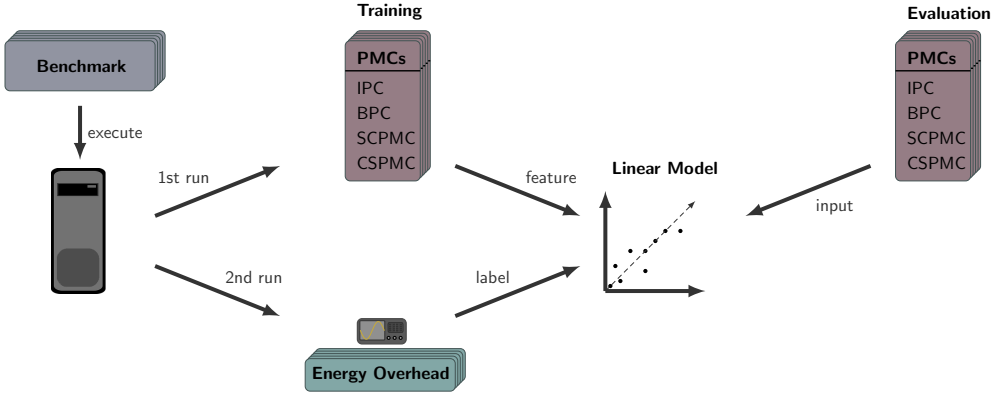
# Energy Prediction Model



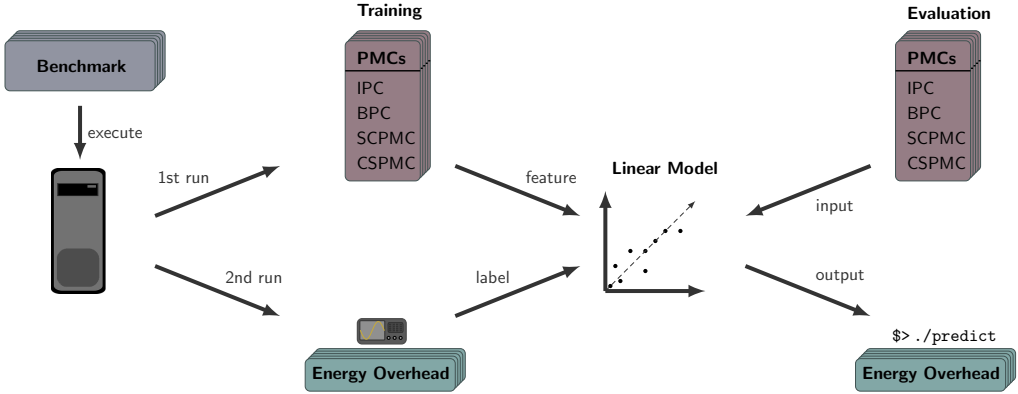
# Energy Prediction Model



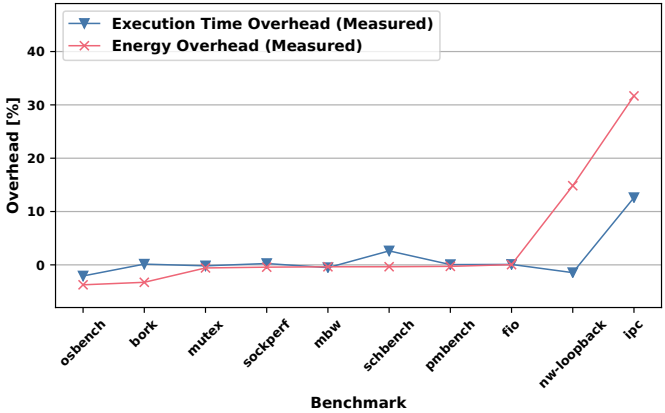
# Energy Prediction Model



# Energy Prediction Model

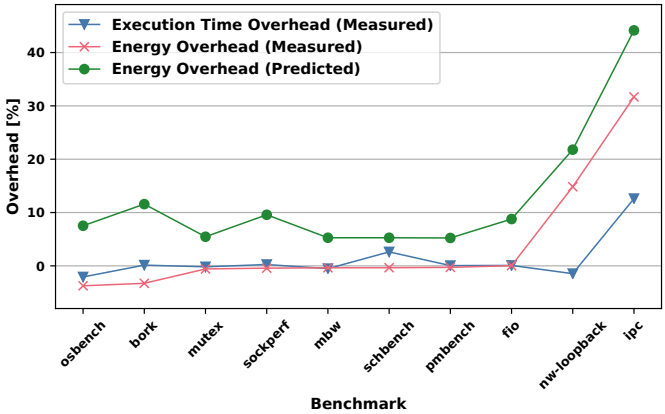


# Q4: Energy Overhead Prediction



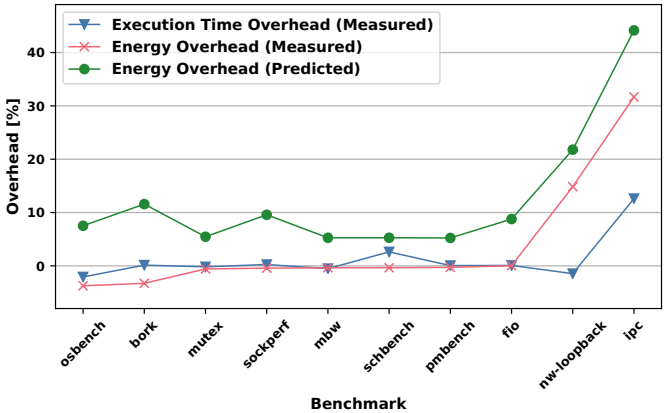

- 10 Phoronix benchmarks
- mostly no overhead
- `nw-loopback`: energy but no time overhead
- `ipc`: energy and time overhead

# Q4: Energy Overhead Prediction




- overestimation of ~5 %
- identifies benchmarks with energy overhead

# Q4: Energy Overhead Prediction




- overestimation of ~5 %
- identifies benchmarks with energy overhead

**Linear model can identify applications with induced energy overheads**

# Performance Counter Correlation

| Performance Counters | Energy Overhead | Time Overhead |
|---|---|---|
| Instructions [per Cycle] | -0.06 | -0.02 |
| Branches [per Cycle] | -0.02 | -0.03 |
| System Calls [per 1M Cycles] | **0.64** | **0.64** |
| Process Context Switches [per 1M Cycles] | **0.41** | **0.33** |

**Spearman Correlation Coefficient**

| | |
|---|---|
| no correlation: | 0.00 |
| **strong correlation:** | **±1.00** |

# Conclusion

**Q1 Energy Overhead?**

→ application dependent (~0 % – 72 %)

→ especially mitigations against Meltdown and Spectre v2

**Q2 Subsystem Related?**

→ operating system interactivity increases overhead

**Q3 Execution Time correlated?**

→ exec. time and energy overhead correlated; exceptions apply

**Q4 Predictable?**

→ applications with overheads predictable

# Conclusion

**Q1 Energy Overhead?**
→ application dependent (~0 % – 72 %)
→ especially mitigations
against Meltdown and Spectre v2

**Q2 Subsystem Related?**
→ operating system interactivity increases overhead

**Q3 Execution Time correlated?**
→ exec. time and energy overhead correlated; exceptions apply

**Q4 Predictable?**
→ applications with overheads predictable

We're Hiring

**Bochum Operating Systems and System Software Group**

CCS center of computer science